

## **Remarks**

The above Amendments and these Remarks are in reply to the Office Action mailed September 18, 2008.

### **I. Summary of Examiner's Rejections**

Prior to the Office Action mailed September 18, 2008, Claims 1-6 and 21-35 were pending in the Application. In the Office Action, Claims 1-6 and 21-35 were rejected under 35 U.S.C. 103(a) as being anticipated by Sarkar et al. (U.S. Patent No. 6,754,659) in view of Kilgore et al. (U.S. Patent Publication No. 2007/0226682), and further in view of Flores et al. (U.S. Patent No. 5,734,837).

### **II. Summary of Applicants' Amendment**

The present Response amends Claims 1, 2, 5, 6, 23, 24, 26, 27, 30, 31 and 33; and adds new Claims 36-40, leaving for the Examiner's present consideration Claims 1-6 and 21-40. Reconsideration of the Application, as amended, is respectfully requested. Applicants respectfully reserve the right to prosecute any originally presented or canceled claims in a continuing or future application.

### **III. Claim Rejections under 35 U.S.C. § 103(a)**

In the Office Action mailed September 18, 2008, Claims 1-6 and 21-35 were rejected under 35 U.S.C. 103(a) as being anticipated by Sarkar et al. (U.S. Patent No. 6,754,659, hereinafter Sarkar) in view of Kilgore et al. (U.S. Patent Publication No. 2007/0226682, hereinafter Kilgore), and further in view of Flores et al. (U.S. Patent No. 5,734,837, hereinafter Flores).

#### **Claim 1**

Claim 1 has been amended to more clearly define the embodiment therein. As amended, Claim 1 defines:

- 1. A system for designing and executing processes, said system comprising:  
an introspection module that generates a catalog of generic components by  
introspecting a set of exposed application programming interfaces (APIs)*

*of a plurality of heterogeneous applications created in different programming languages and transforming a plurality of implementation-specific components of said heterogeneous applications into the generic components of said catalog wherein the catalog contains a series of entries created during introspection of the APIs, each entry representing a generic component which, when invoked, is bound to at least one of the implementation-specific components of said heterogeneous applications upon execution of the processes;*

*a component manager coupled to the introspection module and operable to manage said catalog generated by the introspection module by defining and organizing the generic components in said catalog; and*

*a process designer coupled to the component manager and operable to:*

- select at least one of the generic components from said catalog managed by the component manager; and*
- graphically construct a process definition that includes a series of graphically represented activities linked by one or more transitions wherein at least one activity of said process definition invokes the selected generic component from said catalog;*
- whereby a single process is assembled from existing components of multiple different programming languages, the components being compiled modular routines of the heterogeneous applications that have been previously created in different programming languages; and*
- a repository for storing the process definition; and*
- one or more process engines that execute said process definition to instantiate a process instance, wherein the process instance interacts with the plurality of heterogeneous applications by invoking the generic components in said catalog and wherein the process instance integrates the plurality of heterogeneous applications into the single process by invoking services from the plurality of heterogeneous applications during execution of the activities of said process;*
- wherein during execution of the process instance, the multiple compiled routines of different heterogeneous applications are invoked in order to complete the activities of said process instance.*

As amended, Claim 1 defines a system for designing and executing processes. Each process is assembled from existing components of multiple different programming languages. These components are compiled modular routines of existing applications which have been previously created in different programming languages and/or implementations. Thus when a single process is executed, it will invoke multiple portions of different applications created in multiple different programming languages.

In this manner, the system is used to collect all of the different applications that may be deployed within an organization and use their services as a single defined process. In addition, new processes can be defined using the same existing applications but combining their services in different ways to create completely new and different processes for different business purposes. All of these processes can be managed independently from one another, changed, modified, distributed, accessed over the internet and so on. As such, the system (BPM system) defined in Claim 1 provides an abstraction layer to integrate all of the existing software applications in an organization and use them in new ways and for new different purposes.

To briefly summarize the cited references used in the rejections, the Sarkar reference teaches a method for running Java beans in an EJB environment. To achieve this, EJB support code is generated and then this support code is used to allow the system to run a Java bean in the EJB container. The Kilgore reference teaches an automatic code generator that generated integration source code to allow a local ERP application to access an external ERP application. More specifically Kilgore's automatic code generator automatically creates an adapter that allows the local system to connect to another external system (Kilgore, par. [0035]). Finally, the Flores reference teaches a method for building an application using visual workflows. This is performed by the user creating a visual workflow and then the compiler uses the visual workflow to create the code for the application, based on the workflow (Flores, col. 6 lines 11-20). However, all of these cited references do not disclose the system defined in Claim 1, even if they could all be combined.

To begin with, amended Claim 1 specifically defines that *a single process is assembled from existing components of multiple different programming languages, the components being compiled modular routines of the heterogeneous applications that have been previously created in different programming languages*. This feature is not contemplated by any of the references. In the Office Action, the Kilgore reference was cited as teaching that such business process architecture was known in the pertinent art (Office action, page 3). However, Kilgore merely teaches automatically generating an adapter to connect one system to another system (see "introspector inspects repository... constructs a canonical representation... generates source code that local system integrates into itself as an adapter." (Kilgore, par. [0034]-[0035])). This adapter is a feature that allows one system to communicate with a different system having a different format. This is different from assembling a single process from multiple components

taken from different programming languages (compiled modular routines taken from different applications created in different languages), as defined in amended Claim 1.

It is noted that Applicant's Specification distinguishes the use of adapters from the present embodiments (*see* application-centric integration, pp. 7-9). The use of adapters can be viewed as allowing two different applications to communicate with one another even though they may have different formats or protocols, etc. In contrast, the system defined in Claim 1 specifically constructs new processes from existing components of different applications that have been created in different programming languages. As such, a single new process may invoke different compiled modular routines selected from many applications which have been previously created in different implementations.

Moreover, amended Claim 1 also defines that *during execution of the process instance, the multiple compiled routines of different heterogeneous applications are invoked in order to complete the activities of said process instance*. Thus, one process instance invokes multiple different components selected from different applications. No such functionality is described nor enabled in the cited references. The ability to run Java beans in an EJB environment (Sarkar) is not the same as this feature. Neither is an adapter for allowing one system to communicate with another (Kilgore) or the ability to create an application by drawing workflows (Flores).

In view of the above comments and amendments, Applicants respectfully submit that Claim 1, as amended, is neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

#### **Claims 23 and 30**

Claims 23 and 30, while independently patentable, recite limitations that, similarly to those described above with respect to Claim 1, are not taught, suggested nor otherwise rendered obvious by the cited references. Reconsideration thereof is respectfully requested.

#### **Claims 2-6, 21-22 and 24-29**

Claims 2-6, 21-22 and 24-29 are not addressed separately, but it is respectfully submitted that these claims are allowable as depending from an allowable independent claim, and further in view of the comments provided above. Applicants respectfully submit that Claims 2-6, 21-22

and 24-29 are similarly neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

It is also submitted that these claims also add their own limitations which render them patentable in their own right. Applicants respectfully reserve the right to argue these limitations should it become necessary in the future.

**IV. Additional Amendments**

The present Response hereby adds new dependent Claims 36-40. Support for new Claims 36-40 can be found on pages 40-41, as well as other portions of the Specification as originally filed. Applicant respectfully submits that no new matter is being added and respectfully requests consideration thereof.

**V. Conclusion**

In view of the above amendments and remarks, it is respectfully submitted that all of the claims now pending in the subject patent application should be allowable, and reconsideration thereof is respectfully requested. The Examiner is respectfully requested to telephone the undersigned if he can assist in any way in expediting issuance of a patent.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 06-1325 for any matter in connection with this response, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: December 18, 2008

By: /Justas Geringson/  
Justas Geringson  
Reg. No. 57,033

Customer No.: 23910  
FLIESLER MEYER LLP  
650 California Street, 14<sup>th</sup> Floor  
San Francisco, California 94108  
Telephone: (415) 362-3800  
Fax: (415) 362-2928